



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Adress: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/705,555	11/10/2003	Attila Barta	RSW920030177US1	5498
51016	7590	04/03/2008		
IBM CORP. (RALEIGH SOFTWARE GROUP) c/o Rudolf O Siegesmund Gordon & Rees, LLP 2100 Ross Avenue Suite 2800 DALLAS, TX 75201			EXAMINER	
			WEI, ZHENG	
			ART UNIT	PAPER NUMBER
			2192	
MAIL DATE	DELIVERY MODE			
04/03/2008	PAPER			

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/705,555

Filing Date: November 10, 2003

Appellant(s): BARTA ET AL.

---

Rudolf O. Siegesmund  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 12/31/2007 appealing from the Office action  
mailed 08/10/2007

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

6744450	Zimniewicz et al.	5-2000
6918112	Bourke-Dunphy et al.	11-2000

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.
2. Claims 1-2, 4-13, 15-24 and 26-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zimniewicz (Zimniewicz et al., US 6,744,450 B1) in view of Bourke-Dunphy (Bourke-Dunphy et al., US 6,918,112 B2).

Claim 1:

Zimniewicz discloses a method, system (see for example, Fig.1 and related text) and computer program product (see for example, col.5, lines 28-36) for installing software components, comprising:

- Initiating installation of components (see for example, Fig.3, steps 86, 88, "Load Setup.sdb", "Call UI Manager to Display Startup Screens" and related text);
- selecting a level of detail to be reported regarding the status of components being installed (see for example, Fig.4b, steps 140-142, "Display Scenario Selection Screen", "Receive User Scenario Selection" and related text);

- generating a report regarding status installation progress of the components being installed (see for example, Fig.4c, step 170, "Display Programs Screen" and related text, also see col.11, lines 42-43, "During this process, an installation-progress page is provided 170.");
- displaying that portion of the report identifying the components at the selected level of detail and their corresponding installation progress status (see for example, Fig.4c, step 170, "Display Programs Screen" and related text, also see col.11, lines 42-43, "During this process, an installation-progress page is provided 170."); and
- recording a user's selected preference for a level of granularity in a log (setup.sdb) and when the user participates in a subsequent installation, using the level in the log as default level (see for example, col.6, lines 34-49, "setup database file (setup.sdb)" and related text; also see Fig.4a-4Bsteps 138, 140, 144 and related text).

but does not explicitly disclose the method comprising grouping each component with other components having plurality numbers of dependent components, those components having more dependencies being in a higher group and those components having fewer dependencies being in a lower group. However, Bourke-Dunphy in the same analogous art of system and method to facilitate installation of components discloses grouping dependent components (see for example, col.3, lines 38-40, "defines the interdependencies for the set of components associated with the given installation", also see col.3, lines 42-44,

"each component requires concurrent installation of all higher-level components that connect that component to the base level of the tree" [emphasis added]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to group the interdependent components together for installation. One would have been motivated to do so to ensure proper component dependency as suggested by Bourke-Dunphy (see for example, col.3, lines 48-49, "to ensure proper component dependency")

Claim 2:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses the method comprising accessing a semantic model, the semantic model comprising an indication of a dependency of a component to be installed upon any other component (see for example, Fig.2, element 85, "Dependency Manager" and related text).

Claim 4:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Bourke-Dunphy further discloses wherein:

- each grouping corresponds to a selectable level of detail (see for example, Fig.2, elements 54, 56, 58 "components A, B, C" and element 60 "Group 1"; and related text);

- displaying the portion of the report comprises displaying the components in the group corresponding to the selected level of detail and the components in groups higher than the group corresponding to the selected level of detail (see for example, Fig.4, column 204 "Application Package" and related text).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to further combine Bourke-Dunphy's teachings in Zimniewicz's method of report/display installation information. One would have been motivated to do so to display all of the option and information the user has selected for final approval as suggested by Zimniewicz (see for example, col.11, lines 23-25")

Claim 5:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses, wherein generating the report comprises generating a tree-like structure having at a highest level branch those components to be installed with the greatest number of dependent components and having at a lowest level branch those components to be installed with the fewest number of dependent components (see for example, Fig.6, element 234 and related text, also see col.13, lines 19-40 about tree structure of parent and child components).

Claim 6:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses the method of claim 5, wherein displaying the portion of the report comprises displaying the branches of the tree-like structure corresponding to the selected level of detail and those branches having a higher level (see for example, col.12, lines 54-57, "The UI utilizes a directory tree structure to display components and their sub-components.").

Claim 7:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses, wherein selecting a level of detail comprises accessing a log file (setup database file) for a user, the log file including a user-selected detail level preference (see for example, col.6, lines 42-43, "setup database file (setup.sdb)").

Claim 8:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses, wherein selecting a level of detail comprises accessing a log file for a user (see for example, col.6, lines 42-43, "setup database file (setup.sdb)'), the log file including a detail level generated from past selections by the user (see for example, col.11, lines 24-26, "All of the options and information the user has selected is displayed for final approval 162", also see

Fig.5-Fig.9 and related text. Thus, the user can navigate back to modify any of his or her selection from these screens).

Claim 9:

Zimniewicz and Bourke-Dunphy disclose the method of claim 1, Zimniewicz further discloses, wherein displaying the status of installation progress comprises displaying each of a plurality of status characteristics with a different visual indicator (see for example, Fig.4a, step 130, "Display Welcome Screens", Fig.4b, step 140, "Display Scenario Selection Screen", Fig.4c, step 176, "Display Finish Page" and related text).

Claim 10:

Zimniewicz further discloses the method of claim 9, wherein the different visual indicators comprise different colors (see for example, col.3, lines 45-49, "Such as check marks, color differences, highlight, etc")

Claim 11:

Zimniewicz further discloses the method of claim 9, wherein the status characteristics are selected from the group comprising pending, in progress, successfully completed and error (see for example, col.11, lines 46-50, "the description text in the header of the progress screen update with the each change in install stage").

Claims 12, 13 and 15-22:

Claims 12, 13 and 16-22 are system version (see for example of Zimmiewicz, Fig.1 and related text) for performing the claimed method as in claims 1, 2 and 4-11 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by Zimmiewicz and Bourke-Dunphy. Thus, they also would have been obvious.

Claims 23, 24, 26-33:

Claims 23, 24 and 26-33 are computer program products version of the claimed method, wherein all claimed limitation functions have been addressed in claims 1, 2 and 4-11 above respectively, wherein the method steps can be implemented as computer program and can be practiced and /or stored on a computer operable media (see for example of Zimmiewicz, col.5, lines 28-36). Thus, they also would have been obvious by the teachings of Zimmiewicz and Bourke-Dunphy.

#### **(10) Response to Argument**

A. Claim 1a (Brief pages 7-8):

The Appellant argues that Zimmiewicz reference is silent as to "selecting a level of detail regarding the status of components being installed". Because Zimmniewicz

teaches a "scenario" is merely the selection of components to be installed, much less a "level of detail to be reported".

In response, as an initial matter, it should be noted that Zimniewicz discloses such a level of detail for reporting on [the] installation actions as his title states: "System and Method of Providing Multiple Instillation Actions", and the Abstract states: "A system and method are presented for a suite integration toolkit (SIT) allowing for the provision and display of a rich set of installation actions...These components and sub-components are displayed in a tree structure in the UI with the install action selected and the disk space required displayed. Differential coloring, shading...are all used to enrich the understanding of the installation...to allow the SIT to query the component for its available installation actions and results" [emphasis added].

Zimmniewicz discloses (col.10, lines 48-55) "users can choose 142 from a number of different pre-configured setups, or select custom scenario to completely customize the component or suit installation", (col.9:40-41) "The Setup Manager then installs 90 required baseline components, and calls 92 the UI Manager to display progress information during installation" and Fig.4C step 170 "Display Programs Screen", step 176 "Display Finish Page" and related text description col.11:42-55 "During this process, an installation-progress page is provided 170...the description text in the header of the progress screen updates with each change in install stage. This provides an indication of where in process SIT is... any installation errors encountered during the course of installation are logged 174 to a file specified in the [Error Log] section of the setup data file. A finish page is then displayed 176, and SIT writes out an

unattend file for the completed setup." [emphasis added]. Therefore Zimmniewicz does disclose such limitation in claim1 as argued.

B. Claim 1b (Brief pages 8-9):

The Appellant argues that Bourke-Dunphy is silent to "its tree organization components" and "having the same number of dependent components". Therefore Bourke-Dunphy does not teach the limitation about grouping each component with other components having the same number of dependent components" as cited in claim 1.

In response, first of all, it should be noted that such claimed limitations and argued for "having the same number of dependent components" (emphasis added) is being treated as much as in light of the specification. In that, the originally filed disclosure merely discloses components may be grouped on the basis of the number of components upon which they are dependent [emphasis added] (paragraph [38], lines 5-6), which is reasonable to say that "the same" when or on the basis of match up, agree with, or conform to a number, any number. As Bourke-Dunphy disclosed at col.3, lines 55-64, "the user interface 12 may display a graphical (or textual) indication of a proper dependency for each of the selected components. For example the dependency engine 14 may add other components to user-selected components, such that the user interface 12 displays a composite list of components to the user. The user interface 12 further may present the user with an option to accept the composite list of displayed components (grouping), which conform to predefined dependency rules for the selected components (matched up/agreed with group of components)" [emphasis added].

Therefore, Bourke-Dunphy, indeed, provides a means for grouping on the basis of the number by selectively accepting/adding any number of dependent components including the same number dependent components as argued in claim 1. Moreover, Bourke-Dunphy, does teach “it tree organization components” as in col.3:37-44; “the dependency data 16 may be organized in the form of hierachal tree structure, in which each component requires concurrent installation of all higher-level components that connect that component to the base level of the tree”.

C. Claim 1c (Brief page 9):

The Appellant argues that Zimmniewicz does not exactly disclose the claim limitation about “generating a report regarding status installation progress of the components being installed”

In response, Zimmniewicz discloses generating a report regarding status installation progress by using “an installation –progress page” and “a finish page” as noted above in (A) (see for example, Fig.4C and col.11, lines 42-55). Zimmniewicz further discloses (col.3, lines 45-49) “Additional information is conveyed to the user during the installation process, such as whether or not components are installed already on the computer. This information is presented in various forms, such as check marks, color differences, highlighting etc. (different level of details)” [emphasis added] Therefore, Zimmniewicz does disclose said limitation in claim 1 as argued.

D. Claim 1d (Brief pages 9-10):

The Appellants argues that Zimmniewicz is silent to its program screen "identifying components at the selected level of detail and their corresponding installation progress status".

In response, Zimmniewicz also discloses in Fig.6 about displaying a user interface screen to report installation results by identifying the components at selected level. In Fig.6, under column "Component Name" lists all the components and "+" symbol can be expanded to further display selectable sub-components (components tree). The column "Action" has an "exemplary pull down menu 234 are None (Installed, which results in the check mark being displayed once selected ..." can be used to indicate the selected component's installation result (see for example, Fig.6, element 234 and related description at col.13, lines 1-40, "None (Installed)", "if the component were not installed, the option of Install would replace the action Remove. In this way, only these actions that are available based on the components that are shown to the user") [emphasis added]. Therefore, Zimmniewicz does disclose said limitation in claim 1 as Appellants argued.

E. Claim 1e (Brief page 10):

The Appellant argues that Zimmniewicz does not disclose the limitation about "recording a user's selected preference for a level of granularity in the log and when the user participates in a subsequent installation, using the level in the log as a default level".

In response, it should be noted that the plain language of "recording a user's selected preference for a level of granularity in a log" is not necessary to be interpreted as "the level of detail to be reported". Zimniewicz discloses (col.6, lines 34-49): "a setup database file (setup.sdb) to identify components and their available actions to be performed during the installation and setup hereof" which can be used to save configuration information; also discloses (col.7, lines 27-29) "The UI Manager 91 may also display any customized pages identified in the setup.sdb file during the installation and setup of the suite" [emphasis added]. As noted in (A) above, Zimniewicz also discloses (col.11:42-55) "During this process, an installation-progress page is provided 170...the description text in the header of the progress screen updates with each change in install stage. This provides an indication of where in process SIT is... any installation errors encountered during the course of installation are logged 174 to a file specified in the [Error Log] section of the setup data file. "A finish page is then displayed 176, and SIT writes out an unattend file for the completed setup." [emphasis added]. Thus, it is clear that the customized page (user's selected preference) can be logged (setup.sdb) and used as default during installation. Therefore, the Examiner asserts that Zimniewicz does disclose the feature "user preference log" in amended claim.

#### **(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Zheng Wei/

Wednesday, March 26, 2008

Conferees:

SPE Tuan Q. Dam, Group Art Unit 2192

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192

SPE Eddie C. Lee, Technology Center 2100

/Eddie C Lee/

Supervisory Patent Examiner, TC 2100